OPTIMIZATION OF THE LOADING OF GOODS INTO THE CARGO COMPARTMENT OF A VEHICLE AS A DYNAMIC PROGRAMMING PROBLEM

LAMPA Martin[1], SAMOLEJOVÁ Andrea[1], SIKOROVÁ Andrea[1]

[1]VSB - Technical University of Ostrava, Ostrava, Czech Republic, EU

Abstract

In managerial decision-making, we encounter situations that include simple or complex decisions. The company management often must take into account the sequence of decisions where each decision affects the next one. A tool used in solving these types of subsequent decision-making problems is called dynamic programming.

An integral part of logistics processes is also the loading of goods (products) into the cargo compartment of the vehicle and their shipment to the customer.

This article deals with optimizing the loading of goods into the limited cargo compartment of a vehicle in order to maximize the value of the cargo.

Keywords: Optimization, management, dynamic programming problem

1. INTRODUCTION

There is no simple model for solving dynamic programming problems. Therefore, these problems are classified into groups, each of which has a formulation and method of solution. However, the basic approach and the logic of solving all the problems of dynamic programming is the same. [1]

Dynamic programming can solve problems that can either be segmented into a sequence of decisions or that are composed of a series of small problems to start with. [2]

Analysis in dynamic programming is based on the Bellman's optimality principle, which says: "The optimal method has the property that whatever the initial state and initial decision are, any remaining decisions must create an optimal tactic given the state resulting from the decision."

The understanding of the optimality principle is that if we start in a current stage, the optimal decision for the remaining stages depends only on the state in the current stage and not on the means that brought the system to this state (the optimal tactic is independent of decisions used in earlier stages). [3]

Unlike most other mathematical models for dynamic programming, there is no standard recursive relationship [4]. It is therefore not possible to use a generic calculation tool (e.g. the simplex method of linear programming). It is however possible to classify the problems of dynamic programming into "groups" and build a special calculation process for each of them [5]. Despite the fact that these groups differ in structure and calculation procedures, they use the general approach of dynamic programming. These groups are:

- Allocation problems - these problems are broken into smaller problems,
- Multi-periodic problems - are divided into two or more sections,
- Network problem - PERT and other networks may often be viewed as dynamic programming problems, and so resolved,
- Multiphase problems - these problems arise from situations in industrial production,
- Feedback problems - these problems typically occur in electronics, aerospace, automotive industry,
- Markov decision problems.

2. MATHEMATICAL DESCRIPTION OF DYNAMIC PROGRAMMING

The relationship between the yield at each stage and optimal yield is the key to the process of dynamic programming. [6] The relationship of the individual yields is called a recursive relationship. For dynamic programming, it is important to write recursive relationships for each problem. When the equations are written down, we can perform the dynamic programming calculations. The recursive relationship tells us that the optimal yield in each stage for each of the states is determined as the value of the best option, with each option including the total immediate yield and optimal yield calculated in the previous stage [7].

General description

n - index for the current stage indicates how many stages there are from the current state to the end of the problem,

n-1 - previous stage,

s_n - system state in the current stage to which the recursive relations relate,

s_{n-1} - state in the previous stage,

$f_n(s_n)$ - total yield realized for each variation from state sn in stage n to the end of the problem,

$f^*_n(s_n)$ - optimal total yield, best fn(sn) from state sn in stage *n*,

$f^*_{n-1}(s_{n-1})$ - optimal yield obtained in the previous stage,

$r_n(s_n,d_n)$ - optimal yield realized in stage *n*, when the decision d_n is made for the specific value s_n of the state variable,

d_n - decision between variants made in stage *n* in the currently considered state.

The recursive relationship for state *s* in stage *n* is then:

$$f^*_n(s_n)=min\ d_n[r_n(s_n,d_n) + f^*_{n-1}(s_{n-1})] \quad (1)$$

Dynamic programming reduces a complex problem into a series of simpler problems. After the actual reduction, it is still necessary to solve the subproblems. The subproblems can be solved with methods such as:

- Calculation - in many cases very effective, because the number of possible solutions of subproblems is finite and small,
- Mathematical programming - in many cases the subproblems represent problems of linear or non-linear programming and can be solved as such,
- Sequential search - in some cases iterative procedures may be used in which the solution is improved step-wise.

3. APPLICATION OF DYNAMIC PROGRAMMING TO THE PROBLEM OF STORING GOODS INTO A LIMITED SPACE

We have 4 types of palettes of commodities with a limited delivery quantity for each of them. The weight and value of commodities are given in **Table 1**. Find which items and in what quantity they should be loaded into the compartment of a truck, if its maximum load is 11 tons and the goal is to maximize the value of the cargo.

Table 1 Problem input data

Item	Weight [t]	Value
Palette A	2	18
Palette B	4	25
Palette C	5	30
Palette D	3	20

3.1. Mathematical description

n - stage under consideration,

x_n - number of items of type n to load,

v_n - value of item of type n,

w_n - weight of item of type n,

K - maximum available capacity,

s_n - state (remaining available weight) in stage n.

3.2. Formulation of dynamic programming

This problem belongs to the group of allocation problems.

Stage means any kind of item.

State is the remaining capacity available for allocation, i.e. states 0, 1, 2, 3 …. 11.

At each stage, it is necessary to decide how many units of each item are to be incorporated into the optimal mix.

The following recursive relationship describes this problem:

$$f^*_n(s_n)=min\ x_n[v_n x_n + f^*_{n-1}(s_n-w_n x_n)] \quad (2)$$

where,

s_n - the remaining weight for allocation,

$v_n x_n$ - current yield,

$f^*_n(s_n)$ - optimal total yield in stage n for state s_n,

$f^*_{n-1}(s_n-w_n x_n)$ - optimal yield in the previous stage.

Solutions will involve four stages, because there are 4 items.

Stage 1

Table 2 shows states on the left-hand side. On the top, there are the numbers of palette D that can be loaded.

Table 2 Stage 1. - palette D

State s_1 tons suitable for allocation for palette D	$f_1 s_1 = v_1 x_1 = 20{*}x_1$ (number of palette D to load - 3 tons each)				$f^*_1(s_1)$
	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	
0	0				0
1	0				0
2	0				0
3	0	20			20
4	0	20			20
5	0	20			20
6	0	20	40		40
7	0	20	40		40
8	0	20	40		40
9	0	20	40	60	60
10	0	20	40	60	60
11	0	20	40	60	60

Since the weight of palette D is 3 t, either 0, 1, 2 or 3 units may then be loaded. The table gives the yield calculated according to the formula specified in the table above.

Mathematical description of stage 1:

The yield function is given by the formula $v_1x_1 = 20*x_1$, where x_1 determines the number of units palette D. The column of optimal solution is expressed as $f^*_1(s_1)$ = *maximum* (v_1x_1), where $f^*_1(s_1)$ is the optimal yield starting from state s_1 and using the optimal method from stage 1 to the end.

Stage 2

At this stage, palette C with the remaining weight are allocated based on the best procedure recommended in stage 1. **Table 3** shows yields for different states. Total yields are f2(s2) = 30*x2 + f*1(s2 - 5x2); and optimal yield is max f1(s2).

For instance in state 10, if 10 tons are available, the following 3 alternatives are possible:

- Zero to palette C, then 10 palette D. From **Table 2** we can derive that the optimal allocation of 10 palette D gives a yield of 60,
- One to palette C leaves 5 tons. The best allocation of 5 (from state 1) gives 20 + 30 obtained from allocation 1 to palette C for a total yield of 50.
- One to palette D, two to palette C, takes 10 tons; that means, nothing is left. Therefore, the yield is 2*30 = 60.

Table 3 Stage 2. - palette C

State s_2 tons suitable for allocation of palette C and palette D	$f_2s_2 = 30*x_2 + f*_1(s_2 - 5x_2)$ (number of palette C, each 5 tons)			$f*_2(s_2)$
	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	
0	0			0
1	0			0
2	0			0
3	20			20
4	20			20
5	20	30		30
6	40	30		40
7	40	30		40
8	40	50		50
9	60	50		60
10	60	50	60	60
11	60	70	60	70

Stage 3

The results are listed in **Table 4**. The function of total yield is:

$$f_3(s_3) = 25*x_3 + f^*_2(s_3 - 4x_3) \quad (3)$$

and $f^*_3(s_3) = max\ f_3(s_3)$

Table 4 Stage 3. - palette B

State s_3 tons suitable for allocation of palette C, palette D and palette B	$f_2 s_2 = 25^* x_3 + f^*_2(s_3 - 4x_3)$ (number of palette B, each 4 tons)			$f^*_3(s_3)$
	$x_3 = 0$	$x_3 = 1$	$x_3 = 2$	
0	0			0
1	0			0
2	0			0
3	20			20
4	20	25		25
5	30	25		30
6	40	25		40
7	40	45		45
8	50	45	50	50
9	60	55	50	60
10	60	65	50	65
11	70	65	70	70

Stage 4

The results are listed in **Table 5**. The function of total yield is:

$$f_4(s_4) = 18^* x_4 + f^*_3(s_4 - 2x_4) \quad (4)$$

and $f^*_4(s_4) = max\ f_4(s_4)$

Table 5 Stage 4. - palette A

State s_4 tons suitable for allocation of palette C, palette D, palette B and palette A	$f_4(s_4) = 18^* x_4 + f^*_3(s_4 - 2x_4)$ (number of palette A, each 2 tons)						$f^*_4(s_4)$
	$x_4 = 0$	$x_4 = 1$	$x_4 = 2$	$x_4 = 3$	$x_4 = 4$	$x_4 = 5$	
11	70	78	81	84	**92**	90	**92**

4. OPTIMAL SOLUTION

The initial state is $s_4 = 11$ and its solution is the solution of the whole problem. The optimal solution reads as $x_4 = 4$, i.e. 4 palettes A (weight 8 tons). The remaining $11 - 8 = 3$ tons are allocated in an optimum manner according to the 3rd stage (**Table 4**). Then $x_3 = 0$ and no palette B is loaded. The check then continues to stage 2 (**Table 3**). From table $x_2 = 0$, finally moving to stage 1. (**Table 2**). Optimal solution for 3 tons is $x_4 = 1$.

Therefore, the best solution is:

$x_4 = 4$ palettes A,

$x_1 = 1$ palette D.

Total yield is *18*4 + 20*1 = 92*.

5. CONCLUSION

The allocation problem has several variants. For instance, it may be required that at least one unit (or more than one unit) of each item shall be loaded. The goal may be to minimize costs. Additional restrictions can also be attached, such as limits of volume. Due to the special structure of dynamic programming it is difficult to design a standard computer program for it. Either a special program for every problem must be designed or an extremely large variety of options must be constructed.

ACKNOWLEDGEMENTS

The work was supported by the specific university research of the Ministry of Education, Youth and Sports of the Czech Republic No. SP2016/59.

REFERENCES

[1] TURBAN E., MEREDITH J. R. Fundamentals of management science, USA: R. R. Donnelye & Sons, 1991.

[2] GROS I., DYNTAR J. *Matematické modely pro manažerské rozhodování*, Praha: Vysoká škola chemicko-technologická, 2015.

[3] BESTA P., SAMOLEJOVÁ A., LENORT R., ZAPLETAL F*. INNOVATIVE APPLICATION OF MATHEMATICAL METHODS IN EVALUATION OF ORE RAW MATERIALS FOR PRODUCTION OF IRON,* METALURGIJA, Vol. 53, No. 1, 2014, pp. 93-96.

[4] SUCHÁČEK J., SEĎA P. *Stochastic conception of input-output model: Theoretical and practical aspects,* International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 6, No. 6, 2012, pp. 739-747.

[5] KUTÁČ J., MYNÁŘ M., ŠVECOVÁ E., KOUDELA M. *INTERNAL AUDIT AND CONTROLLING IN METALLURGY*. In METAL 2011: 20th Anniversary International Conference on Metallurgy and Materials. Ostrava: TANGER, 2011, pp. 1323-1327.

[6] SAKÁL P., JERZ V. *Operačná analýza v praxi manažéra*. Trnava: SP SYNERGIA, 2003.

[7] BRITO P. *Indroduction to Dynamic Programming Applied to Economics* [online]. 2008 [cit. 1.10.2016]. Dostupné z: http://www.fep.up.pt/docentes/joao/material/aea/notas_pbrito_2008.pdf